Package 'AdaJoint'

August 8, 2012

Title Adaptive Joint Test

Version 0.0.1

Date 2012-07-10

Author Kai Yu, Han Zhang

Description Powerful gene-based test via variable selection for genome-wide association studies

Maintainer Kai Yu <yuka@mail.nih.gov>

Depends mvtnorm

License GPL-2

Archs i386, x64

R topics documented:

AdaJoint *Powerful gene-based test via variable selection for genome-wide association studies*

Description

An R package for computing gene p-values using the Adaptive Joint Test. This package can be used to analyze genes and pathways based on a genetic association study with a binary case-control outcome.

Details

Single-marker analysis that evaluates one genetic marker a time is the most commonly used approach for the identification of disease susceptibility loci in genome-wide association (GWA) studies. Increasing empirical evidence has suggested that there are regions or genes consisting of multiple genetic variants, which jointly contribute to the disease risk. A gene-based test, which evaluates the association between the outcome and all SNPs in the gene simultaneously, can be a more effective approach than the single-marker approach for studying such regions, thus is helpful to uncover some of missing heritability.

The package contains 3 types of gene-based tests: AdaJoint, AdaJoint2, and ARTP. The AdaJoint test starts the greedy search with the best marginal model with one SNP, hence its performance partially depends on the power of marginal test. An alternative strategy is starting from the best model with two SNPs through an exhaustive search (AdaJoint2). ARTP is the Adaptive Rank Truncated Product test.

The main function is `adajoint` which allows the user to pass in a data frame which contains all the data, or allows the data to be stored in files and have the function read in the data. If the user already has files containing test statistics for each gene, then the function `pathway.pvals` can be called to compute the gene and pathway p-values.

Author(s)

Kai Yu <yuka@mail.nih.gov> and Han Zhang <han.zhang2@nih.gov>

References

Zhang H, Liang F, Wheeler W, Shi J, Yu K, Powerful gene-based test via variable selection for genome-wide association studies, submitted

Yu K, Li Q, Bergen AW, Pfeiffer RM, Rosenberg PS, Caporaso N, Kraft P, Chatterjee N Pathway analysis by adaptive combination of P-values Genet Epidemiol 33(8):700-9; 2009 Dec.

adajoint *Powerful gene-based test via variable selection for genome-wide association studies*

Description

Calculate gene p-values using the Adaptive Joint Test

Usage

```
adajoint(obj, pheno.list, gene.list, op=NULL)
```

Arguments

obj	A data frame containing all variables or a list of type snp.list.
pheno.list	A list describing the covariate and response data. See pheno.list. Note that if obj is a data frame, then pheno.list$file and pheno.list$id.var do not need to be specified.
gene.list	A list describing the SNP-gene-group data or NULL. See gene.list. If NULL, then it is assumed that all SNPs in the genotype file belong to the same gene. If obj is a data frame, then gene.list cannot be NULL.
op	List of options. See details.

Details

If obj is a data frame, then all variables used in the analysis should be numeric. Otherwise if obj is a list, then the genotype data defined by snp.list and phenotype data defined by pheno.list will be merged together into a data frame with the SNPs coded as the number of copies of the minor allele. Missing genotypes for a SNP will be imputed as the mean value using the non-missing genotypes. A random seed should be set before calling pathway.normal in order to reproduce results. The randomness is due to the ranking of p-values, where ties are broken randomly.

SNPs in high LD:
Highly correlated SNPs within a gene can cause numerical problems, so SNP filtering needs to performed in such a case. One way to accomplish this filtering is to set op$filter.R2 to a positive number so that the r2fast() function in the GenABEL package will be called to compute the R^2 values between each pair of SNPs and remove one SNP in each pair with R^2 > op$filter.R2. Another way to filter SNPs is to set the option gene.list$exclude.snps or gene.list$include.snps.

Options list:
Below are the names for the options list op. All names have default values if they are not specified.

- method 1-3: 1=AdaJoint, 2=AdaJoint2, 3=ARTP. The default is 1.
- nperm Number of permutations. The default is 10000.
- out.dir Output directory for temporary files. The default is the working directory getwd
- delete 0 or 1 to delete (temporary) files containing the test statistics for each gene. The default is 1.
- filter.R2 Threshold to filter out SNPs that are in LD within each gene. Set to 0 for no filtering. NOTE: the GenABEL package must be loaded to use this option. The default is 0.

Options for gene-based tests:

- inspect.snp.n The number of candidate truncation points to inspect the top SNPs in a gene. The default is 1.
- inspect.snp.percent A value x between 0 and 1 such that a truncation point will be defined at every x percent of the top SNPs. The default is 0 so that the truncation points will be 1:inspect.snp.n.

Options for pathway-based test:

- `inspect.gene.n` The number of candidate truncation points to inspect the top genes in the pathway. The default is 10.
- `inspect.gene.percent` A value `x` between 0 and 1 such that a truncation point will be defined at every `x` percent of the top genes. The default is 0.05.

Assume the number of SNPs in a gene is 100. Below are examples of the truncation points for different values of `inspect.snp.n` and `inspect.snp.percent`.

inspect.snp.n	inspect.snp.percent	truncation points
1	0	1
1	0.05	5
1	0.25	25
1	1	100
2	0	1, 2
2	0.05	5, 10
2	0.25	25, 50
2	1	100
3	0.2	20, 40, 60

Value

The returned value is a list with names "pathway.pvalue", "gene.table", and "nperm". pathway.pvalue is the Adaptive Rank Truncated Product p-value for the pathway. gene.table is a data frame containing the gene name, number of SNPs in the gene that were included in the analysis, and the p-value for the gene.

References

Zhang H, Liang F, Wheeler W, Shi J, Yu K, Powerful gene-based test via variable selection for genome-wide association studies, to appear

Yu K, Li Q, Bergen AW, Pfeiffer RM, Rosenberg PS, Caporaso N, Kraft P, Chatterjee N Pathway analysis by adaptive combination of P-values Genet Epidemiol 33(8):700-9; 2009 Dec.

See Also

`pathway.pvals`

Examples

```
# Load the sample data
data(data, package="AdaJoint")

# Define pheno.list
pheno.list <- list(response.var="Y", main.vars=c("X1", "X2"))

# Define the gene-SNP list.
# A group variable need not be specified for gene level tests.
gs_file <- system.file("sampleData", "gene_data.txt", package="AdaJoint")
gene.list <- list(file=gs_file, delimiter="\t", header=1,
                  snp.var="SNP", gene.var="Gene")

# Set the options
nperm <- 100
```

```
temp.dir <- "C:/temp/"
op <- list(nperm=nperm, out.dir=temp.dir)

set.seed(123)
# Pass in a data frame to the adajoint function
#adajoint(data, pheno.list, gene.list, op=op)


# Define snp.list
geno_file <- system.file("sampleData", "geno_data.txt", package="AdaJoint")
snp.list <- list(file=geno_file, file.type=2, delimiter="\t")

# Add id.var and file to pheno.list
pheno.list$file <- system.file("sampleData", "pheno_data.txt", package="AdaJoint")
pheno.list$id.var <- "ID"

set.seed(123)
# Keep the data in files and have adajoint read in and merge the data
#adajoint(snp.list, pheno.list, gene.list, op=op)
```

`data` *Sample data*

Description

Sample data for `adajoint`

Details

data.rda is a data frame containing an id variable, response, 2 covariates and 50 SNPs on 500 subjects. The SNPs are coded as the number of copies of the minor allele.

Examples

```
# Load and print a subset
data(data, package="AdaJoint")

data[1:10, 1:10]
```

`gene.list` *List to describe the gene-SNP file*

Description

The list to describe the SNP-gene-group file for `adajoint`

Format

The format is a list:

file Text file containing at least 2 columns, where one column is for the SNPs and the other column is for the gene containing the SNP. No default.

delimiter The delimiter used in `file`.

gene.var Variable name or column number of the gene variable. The default is "Gene".

snp.var Variable name or column number of the SNP variable. The default is "SNP".

header 0 or 1 to denote if `file` contains a header of variable names. The default is 1.

include.genes List of genes to include in the analysis. The default is that all genes will be included.

exclude.genes List of genes to exclude in the analysis. The default is NULL.

include.snps List of SNPs to include in the analysis. The default is that all SNPs will be included.

exclude.snps List of SNPs to exclude in the analysis. The default is NULL.

group.var Variable name or column number of the group variable. This variable is only useful if the user is interested in the pathway p-value (see details). The default is `gene.var`.

Details

All the genes and SNPs listed in this file define a single pathway. SNPs in the same group are considered correlated, so the test statistics for SNPs in the same group will be generated jointly. SNPs belonging to the same gene cannot be broken up into different groups.

`gene_data` *SNP-Gene-Group data*

Description

Sample data file for the `file` argument of `gene.list`

Details

gene_data.txt is a tab delimited file that contains the SNPs within each gene and the genes within each group.

Examples

```
# Load and print the first 5 rows
data(gene_data, package="AdaJoint")

gene_data[1:5, ]
```

geno_data *Sample genotype data*

Description

Sample genotype data file for the `file` argument of `snp.list`

Details

geno_data.rda is a type 1 data file (see `file.type` in `snp.list`). This data contains 50 SNPs and 500 subjects, and is tab delimited. The first row of the data contains the subject ids. Starting from row 2, are the SNP ids and the genotypes for each subject. The genotypes are coded as AA, AG, GG.

Examples

```
# Load and print a substring the first 5 lines
data(geno_data, package="AdaJoint")

substring(geno_data[1:5], 1, 50)
```

pathway.pvals *Compute gene and pathway p-values*

Description

Calculate gene and pathway p-values

Usage

```
pathway.pvals(outfiles, genes, nSNP.gene, op=NULL)
```

Arguments

`outfiles`	Character vector of the names of the files created by `adajoint`
`genes`	Character vector of the gene names corresponding to `outfiles`
`nSNP.gene`	Numeric vector of the number of SNPs in each gene.
`op`	List of options. See `details`.

Details

The input arguments `genes` and `nSNP.gene` are only used in the resulting `gene.table` in the returned object. However, the option `nperm` needs to be correctly set. A random seed should be set before calling `pathway.pvals` in order to reproduce results. The randomness is due to the ranking of p-values, where ties are broken randomly.

Options list:
Below are the names for the options list `op`. All names have default values if they are not specified.

- nperm Number of permutations in each of the outfiles. The default is 10000.
- out.dir Output directory for temporary files. The default is the working directory getwd
- inspect.gene.n The number of candidate truncation points to inspect the top genes in the pathway. The default is 10.
- inspect.gene.percent A value x between 0 and 1 such that a truncation point will be defined at every x percent of the top genes. The default is 0.05.

Value

The returned value is a list with names "pathway.pvalue", "gene.table", and "nperm". pathway.pvalue is the ARTP p-value for the pathway. gene.table is a data frame containing the gene name, number of SNPs in the gene that were included in the analysis, and the p-value for the gene.

See Also

adajoint

---

pheno.list *List to describe the covariate and outcome data*

Description

The list to describe the covariate and outcome data for adajoint

Format

The format is a list:

file Covariate data file. This file must have variable names, one of which being a response variable (see response.var). No default.

id.var Name of the id variable. No default.

response.var Name of the response variable. This variable must be coded as 0 (control) and 1 (case). No default.

main.vars Character vector of variables names for variables in file that will be included in the model as main effects. The default is NULL.

delimiter The delimiter in file. The default is determined from the file.

in.miss Vector of character strings to define the missing values. This option corresponds to the option na.strings in read.table. The default is "NA".

Details

In this list, response.var must be specified. Depending on how the adajoint function is called, file and id.var may also need to be specified. **Missing data:** If any of the variables defined in main.vars or response.var contain missing values, then those subjects will be removed from the analysis.

pheno_data *Sample covariate and response data*

Description

Sample covariate and response data file for the `file` argument of `pheno.list`

Details

The file pheno_data.txt is a tab-delimited type 3 data set (see `file.type` in `pheno.list`). It contains the variables:

- `ID` The subject id
- `Y` Case-control status (0, 1)
- `X1` Continuous covariate
- `X2` Continuous covariate

Examples

```
# Load and print the first 5 rows
data(pheno_data, package="AdaJoint")

pheno_data[1:5, ]
```

snp.list *List to describe the genotype data*

Description

The list to describe the genotype data for `adajoint`

Format

The format is a list:

file File to use. No default.

file.type 2 or 3 (see details).

delimiter The delimiter used in `file`.

in.miss Vector of values to denote the missing values in `file`. The default is " " (2 blank spaces).

heter.codes Vector of codes used for the heterozygous genotype. If NULL, then it is assumed that the heterozygous genotype is of the form "AB", "Aa", "CT", ... etc, ie a 2-character string with different characters (case sensitive). The default is NULL.

id.var (Only for `file.type` = 3) The subject id variable. The default is 1.

Details

In this list, `file` must be specified. If the SNPs are already coded as the number of copies of the minor allele, then set `heter.codes` to 1 (the heterozygous genotype).

`Type 2` has data in the form:

	subject1	subject2	subject3
snp1	AA	CC	AC
snp2	GT	GT	GG

The first row must contain the subject ids. Starting from row 2, the first delimited field must contain the SNP id. The remaining delimited fields contain the genotypes. Rows are SNPs, columns are the subjects.

`Type 3` has data of the form:

id	snp1	snp2
subject1	AA	GT
subject2	CC	GT
subject3	AC	GG

Examples

```
# Suppose the genotype data is a tab-delimited, type 2 file: c:/temp/data/geno1.txt.
#  Also assume the data has the additive coding 0, 1, 2 with NA as missing values.
# The below list is for processing the file.
snp.list <- list(file="C:/temp/data/geno1.txt", delimiter="\t", file.type=2,
                 heter.codes=1, in.miss=NA)
```

Index